

Aware First, Think Less: Dynamic Boundary Self-Awareness Drives Extreme Reasoning Efficiency in Large Language Models

Qiguang Chen^{1*} Dengyun Peng^{1*} Jinhao Liu¹ HuiKang Su¹ Jiannan Guan¹ Libo Qin^{2,✉}
Wanxiang Che^{1,✉}

¹ LARG, Research Center for Social Computing and Interactive Robotics, Harbin Institute of Technology,

² School of Computer Science and Engineering, Central South University

Abstract:

Recent advancements in large language models (LLMs) have greatly improved their capabilities on complex reasoning tasks through Long Chain-of-Thought (CoT). However, this approach often results in substantial redundancy, impairing computational efficiency and causing significant delays in real-time applications. To improve the efficiency, current methods often rely on human-defined difficulty priors, which do not align with the LLM’s self-aware difficulty, leading to inefficiencies. In this paper, we introduce the Dynamic Reasoning-Boundary Self-Awareness Framework (DR. SAF), which enables models to dynamically assess and adjust their reasoning depth in response to problem complexity. DR. SAF integrates three key components: Boundary Self-Awareness Alignment, Adaptive Reward Management, and a Boundary Preservation Mechanism. These components allow models to optimize their reasoning processes, balancing efficiency and accuracy without compromising performance. Our experimental results demonstrate that DR. SAF achieves a 49.27% reduction in total response tokens with minimal loss in accuracy. The framework also delivers a 6.59x gain in token efficiency and a 5x reduction in training time, making it well-suited to resource-limited settings. During extreme training, DR. SAF can even surpass traditional instruction-based models in token efficiency with more than 16% accuracy improvement.

* *Equal Contribution*

✉ *Corresponding Author*



Date: Aug 01, 2025



Code Repository: https://github.com/sfasaffa/DR_SAF



Contact: qgchen@ir.hit.edu.cn, dypeng@ir.hit.edu.cn, car@ir.hit.edu.cn, lbqin@csu.edu.cn

1. Introduction

Recent advancements in large language models (LLMs) have demonstrated their remarkable ability to tackle complex reasoning tasks, particularly with the use of Long Chain-of-Thought (Long CoT) techniques [19, 9, 29]. In contrast to the short chain-of-thought (Short CoT) typically employed in conventional LLMs [51, 41, 7], Long CoT involves a more detailed and progressive process of exploration and reflection based on a given problem. This process is facilitated by inference-time scaling [19, 58, 26]. As a result, Long CoT has significantly advanced areas such as mathematical and logical reasoning. Moreover, it has provided new

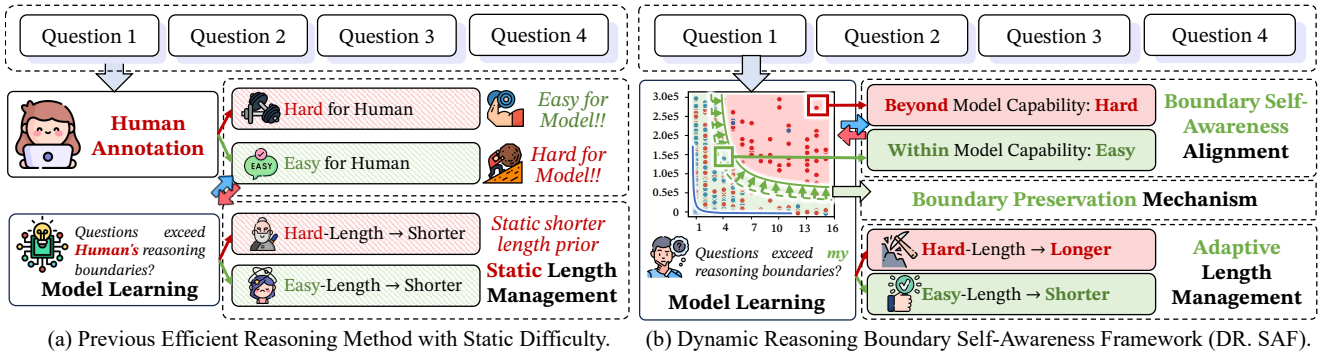


Figure 1: Traditional efficient reasoning training methods (a) primarily determine the difficulty of questions based on human-defined priors, while our dynamic reasoning boundary self-awareness framework (b) judged the difficulty of questions based on model self-awared reasoning boundary.

insights into the role of supervised fine-tuning (SFT) and reinforcement learning (RL) in enhancing the availability of extended reasoning chains [42, 37].

While achieving promising performance, such a Long CoT paradigm generates substantial redundant tokens, significantly impairing computational efficiency and leading to unacceptable application latency [52, 48, 17]. To mitigate this issue, several approaches focus on optimizing reasoning length [50, 47]. Specifically, a series of studies incorporate compression techniques, such as static pruning thresholds to filter out intermediate tokens [36] or adaptive routing to more efficient modules [34, 32, 30]. Other approaches enhance the model’s inherent ability to produce concise reasoning paths through techniques [50, 47]. For instance, AdaptThink [57] and DAST [45] propose frameworks that dynamically adjust models’ reasoning depth based on predefined measures of problem complexity, while Huang et al. [25] extend these paradigms to human-designed adaptive budgeting. However, as illustrated in Figure 1 (a), current methods often rely on manually designed static priors of difficulty and target length, while neglecting each LLM’s intrinsically evolving reasoning boundaries during training [6, 8]. As a result, problems initially identified as “simple” may remain challenging for an LLM whose capability still needs longer exploration, and those previously judged as “complex” might later be handled intuitively by the model by shorter reasoning processes, leading to inefficient reasoning processes and suboptimal performance.

To tackle this challenge, as shown in Figure 1 (b), we present the Dynamic Reasoning-Boundary Self-Awareness Framework (DR. SAF), which assesses problem difficulty relative to a model’s reasoning capacity. Specifically, DR. SAF consists of three key components: (1) **Boundary Self-Awareness Alignment** enables LLMs to recognize their real-time reasoning boundaries. This self-awareness allows the model to assess the difficulty of a given question based on its own capabilities, prompting self-guided adjustments in reasoning depth and answer length. (2) **Adaptive Length Management** further refines efficiency by adapting the reward according to the model’s real-time boundaries. It encourages longer exploration beyond the Completely Infeasible Reasoning Boundary (CIRB) and shorter reasoning within the Completely Feasible Reasoning Boundary (CFRB), ensuring that the model does not oversimplify and compromise quality. (3) **Boundary Preservation Mechanism** maintains stability by preventing the collapse of real-time reasoning boundaries during training, ensuring that all correct responses receive non-negative reinforcement. These innovations address the traditional trade-off issue between efficiency and accuracy, enabling models to dynamically adjust their reasoning depth based on their capabilities.

DR. SAF enhances a model’s boundary self-awareness, enforces boundary-driven length adaptation, and

preserves these boundaries, enabling real-time control of reasoning depth without degrading performance. When evaluated on six public benchmarks, applying DR. SAF to the distilled Qwen-2.5 model reduces total response tokens by 49.27% and achieves state-of-the-art token efficiency. Compared with distilled model, DR. SAF delivers a 6.59x gain in token efficiency. After additional continual training, the extremely compressed DR. SAF model can even surpass traditional instruction-based models in token efficiency and increase accuracy by more than 16%. On the distilled Qwen-3 model, DR. SAF reduces training steps by 80% compared with previous reinforcement-learning-based methods, making it attractive for deployments with limited computational resources.

Our contributions can be summarized as follows:

- We first point out the limitations of existing efficient reasoning methods, which often rely on human-annotated difficulty priors that do not align with LLMs’ reasoning requirements. This misalignment leads to inefficient reasoning processes and suboptimal performance.
- We propose a novel DR. SAF framework, which enables models to dynamically assess their own reasoning boundaries, adaptively manage length reward signals based on problem feasibility, and prevent models from reasoning boundary collapse.
- We systematically demonstrate the effectiveness of DR. SAF through extensive experiments across 6 benchmarks, revealing substantial improvements in efficiency. The extreme speedup can even enable LLMs to surpass the token efficiency of instruction models, while maintaining a 16% improvement in accuracy.

2. Preliminaries

2.1. The Efficient Reasoning Objective

Given an input x , an LLM generates an output $y = \{S, a\}$, which consists of a reasoning step trajectory $S = (s_1, s_2, \dots, s_T)$ and a final answer a .

The objective of efficient reasoning is to develop a policy that minimizes the reasoning path length while preserving accuracy. Formally, the efficient reward is expressed as:

$$R_{\text{Eff}}(y|x) = R_{\text{Acc}}(y|x) + \gamma R_{\text{Len}}(y|x), \quad (1)$$

where $R_{\text{Acc}}(y|x)$ provides a reward of 1 when y is correct, $R_{\text{Len}}(y|x) \propto -\ell_y$ is the length reward, which is negatively correlated with the response length ℓ_y , and γ is a constant hyperparameter.

2.2. Group Relative Policy Optimization

We utilize Group Relative Policy Optimization (GRPO) to optimize LLMs, an efficient, critic-free reinforcement learning method that reduces memory and computational costs. Specifically, GRPO operates through group-wise advantage estimation. For a given input, the policy model π_θ generates a group of k outputs, $\mathcal{Y} = \{y_1, \dots, y_k\}$, evaluated by a reward function to yield reward group $\mathcal{R}_{\text{Eff}} = \{R_{\text{Eff}}(y_i|x)\}_{i=1}^k$. The advantage for each output is computed by normalizing its reward against the group’s statistics:

$$\mathcal{A}(\mathcal{Y}|x) = \frac{\mathcal{R}_{\text{Eff}} - \mu_{\mathcal{R}}}{\sigma_{\mathcal{R}} + \epsilon}, \quad (2)$$

where $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ denote the mean and standard deviation of the rewards group \mathcal{R}_{Eff} , respectively. ϵ is a small constant introduced to prevent division by zero. The policy is refined by minimizing the GRPO loss $\mathcal{L}_{\text{GRPO}}$, which amplifies actions with high advantage and penalizes those with low advantage.

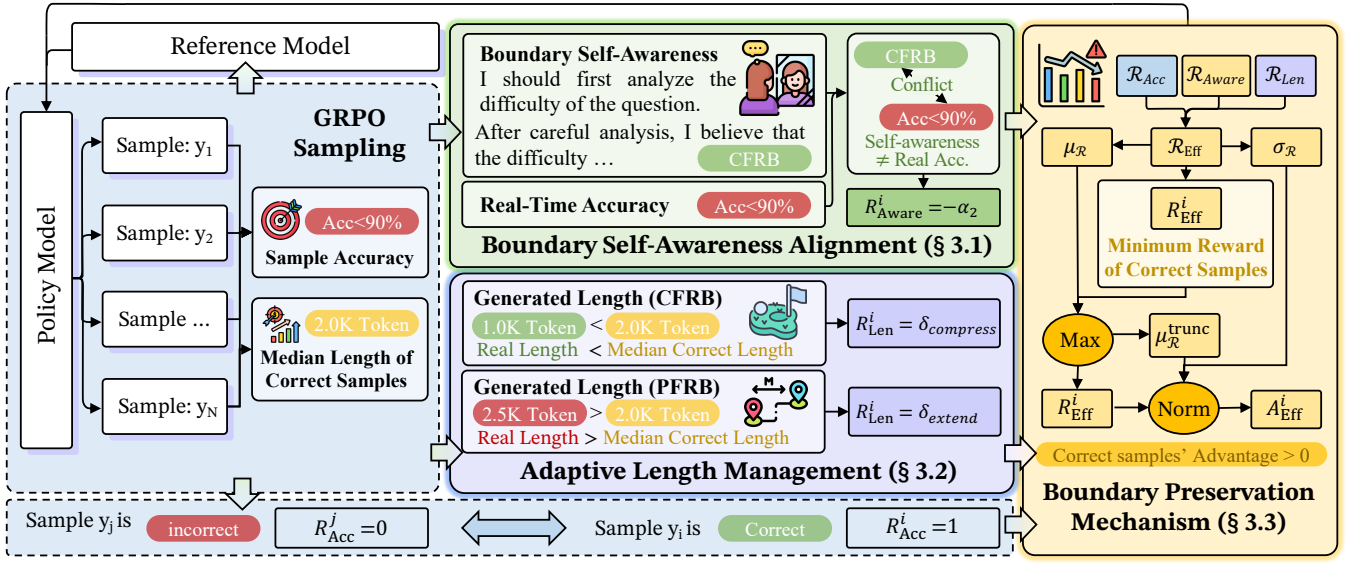


Figure 2: Main pipeline of Dynamic Reasoning-Boundary Self-Awareness Framework (DR. SAF), including Boundary Self-Awareness Alignment (BSA), Adaptive Length Management (ALM), and Boundary Preservation Mechanism (BPM).

3. Methodology

To compress efficiently without sacrificing accuracy, we introduce a three-module framework (see Figure 2): (1) Boundary Self-Awareness Alignment enables model to gauge question difficulty. (2) Adaptive Length Management applies a discrete length-reward schedule that scales with difficulty and reasoning bounds, preventing harmful over-compression. (3) Boundary Preservation Mechanism stabilizes optimization through advantage reshaping. Together, these modules, collectively called DR. SAF, offer a theory-guided solution to balance efficiency and accuracy. Formal proofs of each module’s effectiveness are in the Appendix B.

3.1. Boundary Self-Awareness Alignment

First, the model develops self-awareness of whether a given task falls within its reasoning capabilities. As shown in the green area of Figure 2, the model calibrates perceived task difficulty against its real-time accuracy; any gap between expected and observed performance incurs a reward penalty. Specifically, we employ Boundary Self-Awareness Alignment (BSA) with a format-based reward. Guided by carefully designed prompts, the model evaluates its proficiency on each problem: Inspired by Chen et al. [6], if it determines a problem to be fully mastered, it classifies it as within its Completely Feasible Reasoning Boundary (CFRB) and provides more concise solution; otherwise, it assigns the problem to its Partially Feasible Reasoning Boundary (PFRB), initiating a deeper reasoning process.

Next, to enable adaptive boundary awareness, BSA assesses the model’s accuracy across multiple runs of the same problem. Following Chen et al. [6], we label problems with accuracy above 90% as CFRB and those below 90% as PFRB. When the model’s boundary classification aligns with the problem’s true difficulty, resulting in a correct CFRB or PFRB judgment followed by a correct answer, it receives a positive reward. Conversely, if it mislabels a PFRB problem as CFRB and then answers incorrectly, it incurs a negative reward.

Formally, the reward function is defined as:

$$R_{\text{Aware}}(y|x) = \begin{cases} +\alpha_1, & \text{if } \text{Acc}(\mathcal{Y}|x) \geq 90\% \wedge \text{Aware}(x) < \text{CFRB}; \\ +\alpha_1, & \text{if } \text{Acc}(\mathcal{Y}|x) < 90\% \wedge \text{CFRB} \leq \text{Aware}(x) \leq \text{PFRB}; \\ -\alpha_2, & \text{otherwise,} \end{cases} \quad (3)$$

where α_1 and α_2 are positive constants that scale the rewards, $\text{Acc}(\mathcal{Y}|x)$ is the model’s empirical accuracy on input x , and $\text{Aware}(x)$ denotes the model’s self-assessed difficulty for x . This framework continuously calibrates the model’s self-awareness of reasoning boundaries based on performance feedback.

3.2. Adaptive Length Management

Unlike traditional compression tasks, which focus on unified length penalties, Adaptive Length Management (ALM) introduces staged incentives to generate suitable response lengths. As illustrated in the purple area of Figure 2, tasks in CFRB that LLM already masters receive compression rewards, driving concise reasoning. For low-accuracy tasks beyond Completely Infeasible Reasoning Boundary (CIRB), we give extension rewards to longer incorrect answers, prompting the model to elaborate for deeper exploration.

Specifically, based on k sampling results, we select a correct sample set \mathcal{C} . First, we determine the minimum number of tokens required to maintain a question within the CFRB. Formally, we define $\bar{\ell}_{\text{CFRB}}$, the median response length for correct samples in \mathcal{C} , as the model’s required length for mastery within CFRB. Based on this, we then define two reward types: (1) the **compression reward** δ_{comp} for fully mastered questions. It rewards answers under CFRB whose length ℓ should be below the CFRB mean $\bar{\ell}_{\text{CFRB}}$. (2) the **extension reward** δ_{ext} for exploration-needed questions, those beyond CIRB. Here the answer length ℓ should exceed $\bar{\ell}_{\text{CFRB}}$ (if no correct sample exists, the length threshold will degrade to the average length of all samples $\bar{\ell}_{\text{All}}$). Formally, the adaptive reward for ALM is:

$$R_{\text{Len}}(y|x) = \begin{cases} \delta_{\text{comp}} & \text{if } \text{Acc}(\mathcal{Y}|x) > 90\% \wedge \ell \leq \bar{\ell}_{\text{CFRB}} \\ \delta_{\text{ext}} & \text{if } \text{Acc}(\mathcal{Y}|x) < 10\% \wedge \ell > \bar{\ell}_{\text{CFRB}} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $\delta_{\text{comp}} < R_{\text{Acc}}$ and $\delta_{\text{ext}} < R_{\text{Acc}}$ are positive constants smaller than the accuracy reward R_{Acc} .

3.3. Boundary Preservation Mechanism

Training models with only length penalties and correctness rewards often leads to a common issue known as boundary collapse. In this case, the model excessively compresses reasoning chains, causing the advantage of correct responses to fall below zero. As a result, valid reasoning boundaries collapse, and infeasible ones arise, both undermining performance and destabilizing training. As illustrated in yellow part of Figure 2, we address this by enforcing non-negative advantages for all correct responses.

Let \mathcal{C} represent the set of correct responses, which defines the feasible region. This region includes responses that meet all correctness, length, and awareness criteria within the CFRB, as well as correct responses that may violate secondary preferences (such as length) within the PFRB. Both categories are considered feasible, ensuring that every output $y_i \in \mathcal{C}$ receives a non-negative advantage. Formally, for a given input x , we sample a group of k outputs, $\mathcal{Y} = \{y_1, \dots, y_k\}$, and calculate the total efficient rewards $R_{\text{Eff}} = \{R_{\text{Eff}}(y_i|x)\}_{i=1}^k$ as follows:

$$R_{\text{Eff}}(y_i|x) = R_{\text{Acc}}(y_i|x) + R_{\text{Len}}(y_i|x) + R_{\text{Aware}}(y_i|x). \quad (5)$$

The Boundary Preservation Mechanism (BPM) ensures that correct responses, regardless of length, are not unduly suppressed, thereby preventing boundary collapse. Specifically, we utilize the efficient reward function and the method for calculating boundary preservation advantages. To achieve this, truncated-mean normalization is applied to the output sample group. The advantages of boundary preservation mechanism A are computed as follows:

$$\mu_{\mathcal{R}}^{\text{trunc}} = \max(\mu_{\mathcal{R}}, \min_{y_i \in \mathcal{C}} R_{\text{Eff}}(y_i|x)), \quad (6)$$

where \mathcal{C} is the set of correct responses. This step ensures that the decision boundary for correct responses is preserved, preventing the model from assigning negative advantages to correct but length-variant answers. Next, the boundary preservation advantages are computed as:

$$\mathcal{A}_{\text{Pre}}(\mathcal{Y}|x) = \frac{\mathcal{R}_{\text{Eff}} - \mu_{\mathcal{R}}^{\text{trunc}}}{\sigma_{\mathcal{R}} + \epsilon}, \quad (7)$$

where $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ are the untruncated mean and standard deviation of reward group \mathcal{R}_{Eff} . By bounding the group mean as $\mu_{\mathcal{R}}^{\text{trunc}}$, we ensure:

$$\forall y_i \in \mathcal{C} : \quad \mathcal{A}_{\text{Pre}}(y_i|x) = \frac{R_{\text{Eff}}(y_i|x) - \mu_{\mathcal{R}}^{\text{trunc}}}{\sigma_{\mathcal{R}} + \epsilon} \geq 0. \quad (8)$$

This safeguard guarantees that correct responses, regardless of length variation, always receive a non-negative advantage. By enforcing this, the Boundary Preservation Mechanism ensures that valid outputs never receive suppressed advantages, thus preventing boundary collapse.

4. Experiments

4.1. Experimental Setup

We utilize verl [46] as the reinforcement learning framework on 8 A100-80G GPUs. We randomly sample 5,000 instances from the DeepMath103K [21] as training set, and trained DR. SAF based on two LLMs, R1-distill-Qwen-2.5-7B [19] and R1-distill-Qwen-3-8B [19]. We validate the effectiveness of strategies on AIME24 [1], GSM8K [12], Math-500 [31], AMC23 [3], OlympiadBench [20], and AIME25 [2]. We report three metrics: Accuracy (ACC in %), average response token length (LEN), and token efficiency (EFF). The Token Efficiency (EFF) is defined as the ratio of Accuracy to Length (EFF = ACC / LEN in %), serving as an indicator of the correctness and reasoning efficiency trade-off.

For comprehensive comparison, we adopt three representative paradigms as baselines for efficient reasoning: (1) **Prompting Strategies:** Dynasor-CoT [18] and DEER [55] activate early-exit mechanisms during reasoning; ThinkSwitcher [30] trains a switcher to dynamically choose between long and short CoT. (2) **Offline Strategies:** OverThink [11] fine-tunes on the shortest generated answers. Spirit [13] and ConCISE-SimPO [40] prune tokens based on confidence scores via supervised fine-tuning or direct preference training. AdaptThink [57] and DAST [45] incorporate human-defined difficulty priors to learn efficient reasoning trajectories. **Online Strategies:** Length-Penalty [5] encourages compress all outputs; FEDH [32] applies a human-defined length prior to promote concise reasoning process. Additionally, we compare the token efficiency of instruction models, like Qwen2.5-Ins [53] and Qwen2.5-Math [54].

4.2. Experimental Results

Offline methods yield high accuracy but are less token-efficient than online methods. Offline approaches use gold-standard reasoning trajectories, raising accuracy by more than 6% on AIME24 (see in

Model Name	GSM8K			MATH500			AIME24			AMC			OlymBench			AIME25		
	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF
Qwen2.5-7B-Ins	90.9	279	32.58	74.2	567	13.09	12.0	1016	1.18	47.5	801	5.93	39.2	827	4.74	7.6	1240	0.61
Qwen2.5-7B-Math	93.2	439	21.23	63.4	740	8.57	19.0	1429	1.33	62.5	1022	6.12	31.5	1037	3.04	4.0	2562	0.16
Qwen2.5-7B-Math-Ins	95.2	323	29.47	81.4	670	12.15	10.3	1363	0.76	60.0	1029	5.83	38.9	1027	3.79	9.3	2087	0.45
R1-Distill-Qwen2.5-7B	92.4	1833	5.04	90.8	3854	2.36	49.2	10200	0.48	90.0	6476	1.39	66.1	7789	0.85	35.0	10518	0.33
+ ThinkSwitcher	92.5	1389	6.66	91.3	3495	2.61	48.3	7936	0.61	–	–	–	57.0	5147	1.11	37.5	6955	0.54
+ Dynasor-CoT	89.6	1285	6.97	89.4	2661	3.36	46.7	12695	0.37	85.0	5980	1.42	–	–	–	–	–	–
+ DEER	90.6	917	9.88	89.8	2143	4.19	49.2	9839	0.50	85.0	4451	1.91	–	–	–	–	–	–
+ OverThink	91.4	879	10.39	92.9	2405	3.86	50.0	9603	0.52	–	–	–	–	–	–	–	–	–
+ Spirit	87.2	687	12.68	90.8	1765	5.14	38.3	6926	0.55	–	–	–	–	–	–	–	–	–
+ ConCISE-SimPO	92.1	715	12.88	91.0	1945	4.68	48.3	7745	0.62	–	–	–	–	–	–	–	–	–
+ DAST	86.7	459	18.89	89.6	2162	4.14	45.6	7578	0.60	–	–	–	–	–	–	–	–	–
+ AdaptThink	91.0	309	29.45	92.0	1875	4.91	55.6	8599	0.65	85.0*	4265*	1.99*	58.4*	5988*	0.98*	38.3*	10380*	0.37*
+ Length-Penalty	87.2	263	33.16	89.1	2121	4.20	51.9	7464	0.70	82.5	4411	1.87	59.8	4919	1.22	33.3	8902	0.37
+ FEDH	90.1	218	41.33	88.5	1306	6.50	42.3	7242	0.58	–	–	–	–	–	–	–	–	–
+ DR. SAF	88.1	162	54.38	88.3	1061	8.32	50.6	6288	0.80	90.0	3096	2.91	59.4	3259	1.82	38.2	6764	0.56
R1-Distill-Qwen3-8B	94.2	2135	4.41	90.6	7051	1.28	67.9	20155	0.34	83.5	11931	0.70	60.1	12895	0.47	62.9	20992	0.30
+ FEDH*	94.4	2014	4.69	92.6	6761	1.37	61.3	13463	0.42	94.7	11928	0.79	63.5	12353	0.51	46.7	14730	0.32
+ Length-Penalty*	93.3	604	15.45	92.4	2581	3.58	63.7	12303	0.52	89.2	6166	1.45	68.4	7383	0.93	54.7	12446	0.44
+ DR. SAF	92.3	521	17.72	93.3	2168	4.30	66.0	9807	0.67	95.6	4003	2.39	71.3	5766	1.24	57.9	10692	0.54

Table 1: Performance comparison across mathematical benchmarks. **Bold** marks the best baseline score per metric. For each method we report its most token-efficient variant. Here, “ ” : prompting strategies, “ ” : offline strategies, “ ” : online strategies. Rows are ordered by token efficiency on GSM8K. “*” indicates results reproduced in this study.

Table 1); however, their longer reasoning chains increase token consumption. In contrast, online methods are more economical, forcing fewer tokens and shorter reasoning paths while maintaining competitive accuracy. Thus, whereas offline methods maximize precision, online methods achieve a superior balance between accuracy and token efficiency.

DR. SAF performs state-of-the-art token efficiency with minimal accuracy degradation. We next report the main results on token efficiency, overall efficiency, and accuracy. As shown in Table 1, DR. SAF demonstrates superior performance in reasoning length and token efficiency. DR. SAF reduces the average response token count by 26.53% compared with Length Penalty. The gains are most pronounced on GSM8K with all data within CFRB, where DR. SAF delivers over **90%** shorter reasoning and nearly **10x** higher token efficiency than the distilled backbone.

DR. SAF markedly increases LLM token efficiency relative to static difficulty-based reasoning. As shown in Table 1, DR. SAF is evaluated against two representative static baselines, AdaptThink and DAST, whose difficulty and length are fixed based on human prior. Because these baselines cannot adapt to variations in complexity relative to the model’s capabilities, they often allocate more tokens than necessary. In contrast, DR. SAF dynamically updates its efficiency during reasoning, reducing the average token count by 34.33%. On GSM8K in particular, it achieves a token-efficiency rate of 54.38%, outperforming AdaptThink by more than 20%. This adaptive mechanism consistently delivers higher token efficiency across all benchmarks.

DR. SAF shows significant performance improvements on stronger LLMs. As shown in Table 1, compared to R1-Distill-Qwen3-8B, DR. SAF achieves comprehensive token efficiency gains: on GSM8K, token efficiency improves from 4.41 to 17.72 (**302%** increase) with minimal accuracy trade-off; on MATH500, both accuracy (90.6% to 93.3%) and token efficiency (1.28 to 4.30, **236%** improvement) increase; on AMC, accuracy rises from 83.5% to 95.6% while token efficiency improves by **241%**. These results demonstrate that DR. SAF achieves an excellent balance between computational efficiency and performance.

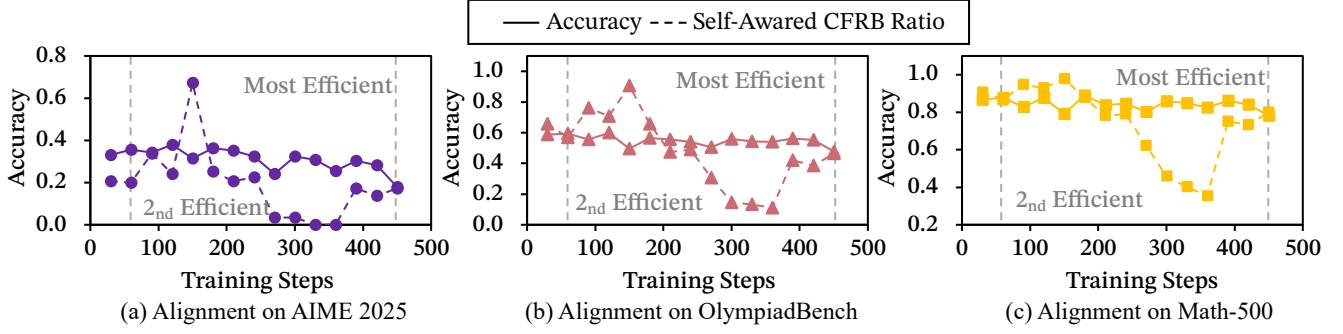


Figure 3: Training trajectory of BSA, shown as the predicted CFRB ratio plotted against the training steps.

4.3. Feature Analysis of DR. SAF

In this section, we analyze the key feature of DR. SAF by addressing three central questions: (1) Can DR. SAF outperform its original instruction backbone in token efficiency? (2) Does DR. SAF offer improved training speed? (3) Does DR. SAF focus solely on reasoning compression without enhancing overall performance?

Answer1: DR. SAF can achieve comparable, or even superior, token efficiency to traditional instruction models across all benchmarks. As shown in Figure 4, earlier techniques match instruction models only on simple datasets such as GSM8K, falling short by over 40% on more complex tasks. In contrast, the further compressed variant DR. SAF-Ext consistently maintains, and even exceeds, the token efficiency of instruction models across varying task complexities. On the CFRB benchmarks (GSM8K and MATH-500), it doubles the token efficiency of prior methods while matching instruction models. On average, DR. SAF improves token efficiency by 211% over previous SOTA approaches and increases accuracy by 16.15% relative to their instruction variance.

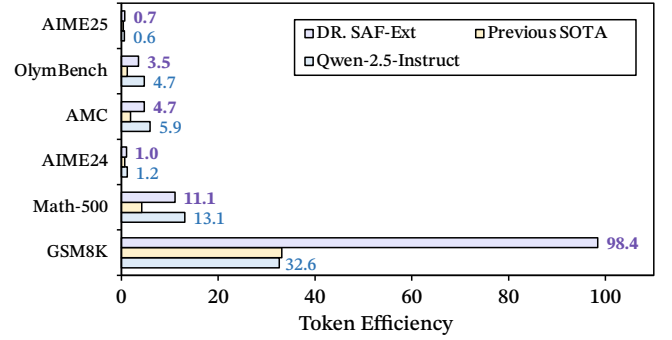


Figure 4: Comparing the extreme efficiency of DR. SAF (DR. SAF-Ext) with traditional instruction models and current SOTA reasoning efficient techniques.

Answer2: DR. SAF achieves significant compression training speedup. Compared to previous RL methods, as shown in Figure 5, DR. SAF reduces training time by up to 5-6 times while maintaining high efficiency. This speedup is particularly evident in large-scale datasets, where DR. SAF minimizes the computational cost associated with model training. Benchmarks indicate that DR. SAF’s compression strategy not only enhances training speed but also ensures minimal loss in model performance, with accuracy even improved. These results demonstrate DR. SAF’s ability to achieve fast and efficient training, mak-

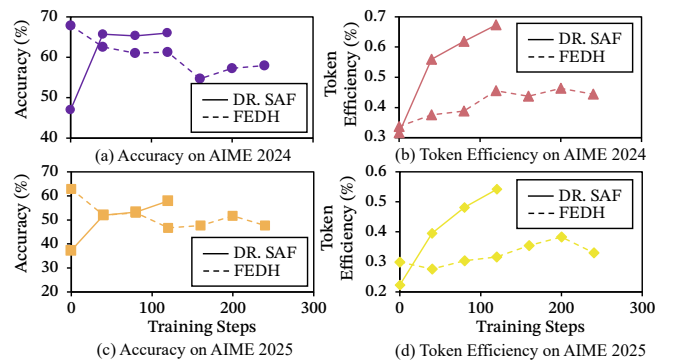


Figure 5: Training efficiency comparison of DR. SAF vs. FEDH on R1-Distill-Qwen-3-8B.

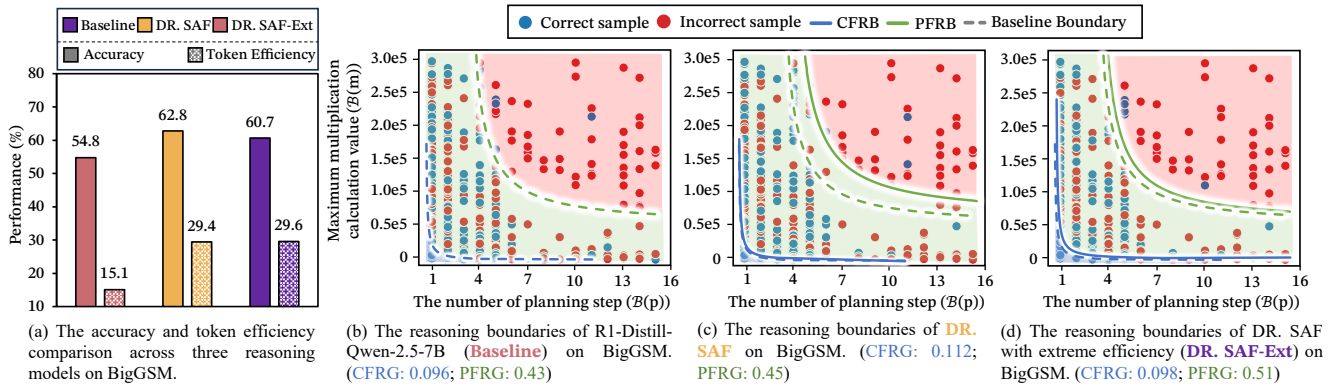


Figure 6: Boundary Preservation Mechanism’s impact on reasoning boundaries on BigGSM [6].

ing it highly scalable for practical applications.

Answer3: DR. SAF achieves performance improvement during compression while length penalty-based methods decrease the performance. In contrast to length penalty-based methods, which often lead to performance degradation during compression, as shown in Figure 5 (a,c), DR. SAF demonstrates a unique ability to enhance model performance even as it compresses reasoning length. As a result, DR. SAF not only maintains high accuracy but also improves it in many cases.

4.4. Module Effect Analysis of DR. SAF

This section evaluates the key effects of each module in DR. SAF with three central questions: (1) Does BSA enhance the model’s boundary self-awareness, thereby improving token efficiency? (2) Does ALM adaptively control token length to simultaneously improve accuracy and efficiency? (3) Does BPM prevent reasoning boundary collapse during compression training, thus preserving model performance?

Answer1: Boundary Self-Awareness Alignment is crucial for DR. SAF efficiency. We assess the effectiveness of the Boundary Self-Awareness Mechanism by ablating it from the DR. SAF. As shown in Table 2, token efficiency decreases by more than 40% without this component. Further, Figure 3 reveals that, during training, the model’s predicted task difficulty progressively aligns with its actual reasoning accuracy. Notably, the models with the highest and second-highest alignment scores also achieve the highest and second-highest levels of token efficiency, respectively.

Model Name	ACC _{AVG}	LEN _{AVG}	EFF _{AVG}
DR. SAF	75.28	2773.2	13.65
w/o BSA	74.98	3219.9	8.04
w/o ALM	67.54	2105.1	11.96
w/o BPM	67.87	2543.6	13.65

Table 2: Ablation analysis of the model’s average accuracy, length, and efficiency scores across GSM8K, MATH500, AIME24, AMC, and OlymBench.

Answer2: Adaptive Length Management is crucial for both accuracy and efficiency in DR. SAF. Replacing Adaptive Length Management with a simple length penalty lowers average response length by 7.74% and token efficiency score by 1.69 (Table 2). Figure 7 further shows that, although the penalty initially shortens reasoning in CFRB (GSM8K), efficiency declines once responses fall below a critical threshold or when tackling harder problems such as AIME. These results confirm that ALM is necessary to sustain the optimal trade-off between brevity and efficiency.

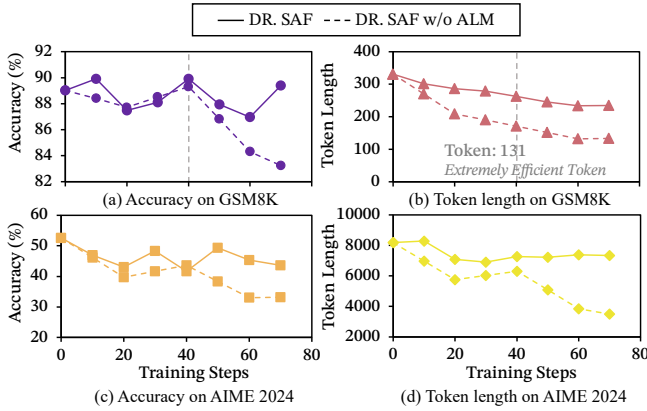


Figure 7: Trends in accuracy and response length during training with Adaptive Length Management.

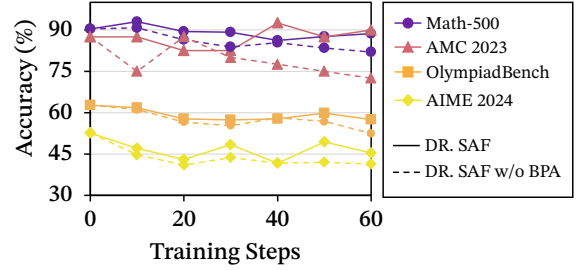


Figure 8: Accuracy trend produced during training by the boundary-preservation mechanism.

Answer3: The Boundary Preservation Mechanism effectively mitigates the reasoning boundary collapse of DR. SAF. As indicated by the ablation results in Table 2, removing this module leads to a 7.41% decrease in accuracy, while token efficiency remains unchanged. This suggests that BPM primarily preserves the model’s reasoning boundaries rather than enhancing token compression. The preservation effect is further supported by the results in Figure 6, where BPM raises Out-of-Domain accuracy on BigGSM from 54.8% to over 60.7%. Notably, DR. SAF maintains improved performance even under extreme token compression scenarios (DR. SAF-Ext), highlighting BPM’s robustness across varying compression levels. Consequently, as depicted in Figure 8, BPM effectively reduces the extent of boundary collapse in DR. SAF, thereby preventing significant performance degradation across a diverse range of tasks and benchmarks.

5. Related work

Long Chain-of-Thought (CoT) prompting has substantially advanced domains such as mathematical and logical reasoning [23, 22, 27, 24]. Furthermore, it has offered novel insights into the contributions of supervised fine-tuning (SFT) and reinforcement learning (RL) for improving both the acquisition and exploration of extended reasoning chains [42, 37, 9]. Notably, confidence-aware methods let LLMs spend compute only when needed. During reasoning, techniques such as prolonged reasoning [34] and dynamic early exit [55] utilize output probabilities to decide when to stop reasoning [40]. Eo et al. [16] uses evaluators for dynamic stopping, and [15] exploits reasoning structure confidence. Additionally, Length-filtered Vote [52] filters reasoning sequences based on confidence-length correlations. Other approaches make pre-reasoning decisions. HybridLLM [14], RouteLLM [38], System-1.x [44], and DynaThink [39] train confidence-based triggers to determine reasoning necessity [35, 59, 28]. AdaptThink [57] optimizes constrained objectives to encourage “no-thinking” modes for human-defined easy questions [25]. Early efficiency-oriented RL approaches relied on direct length penalties for model outputs [33, 49, 56, 36, 32]. More recent work introduces staged training and adaptive rewards: multi-phase and meta-RL frameworks re-allocate compute and token budgets in real time [50, 47, 43]. By linking CoT length to problem difficulty, newer methods keep explanations brief on easy tasks and detailed on hard ones, reducing latency without harming accuracy. Specifically, An et al. [4] imposes a human-defined token budget that grows with task complexity, while Ling et al. [32] embeds length penalties during training to balance brevity and reasoning depth based on human priors, jointly cutting inference time.

However, traditional efficiency-target methods predominantly focus on optimizing reasoning paths based

on fixed or human-defined difficulty levels. In contrast, DR.SAF introduces a self-aware system capable of dynamically adjusting the depth of reasoning according to the model’s internal capabilities and the real-time complexity of the task. This approach enhances both efficiency and accuracy.

6. Conclusion

In this work, we introduce the Dynamic Reasoning-Boundary Self-Awareness Framework (DR.SAF) to incorporate a self-aware system that adjusts the reasoning depth according to the model’s internal capabilities and real-time task complexity, thereby improving both efficiency and accuracy. Extensive experiments on benchmarks such as Math-500 and AIME demonstrate that DR.SAF cuts response tokens 50% and trains 5x faster than long chain-of-thought baselines, yet keeps top-tier accuracy across 6 benchmarks. This framework sets the foundation for more scalable, efficient, and reliable LLMs in real-world applications, balancing reasoning depth with performance.

References

- [1] AIME. American invitational mathematics examination (aime) aime 2024-i & ii, 2024. URL https://huggingface.co/datasets/Maxwell-Jia/AIME_2024.
- [2] AIME. American invitational mathematics examination (aime) 2025-i & ii, 2025. URL <https://huggingface.co/datasets/opencompass/AIME2025>.
- [3] AMC. American mathematics competitions, 2023. URL https://artofproblemsolving.com/wiki/index.php/AMC_Problems_and_Solutions.
- [4] Sohyun An, Ruochen Wang, Tianyi Zhou, and Cho-Jui Hsieh. Don’t think longer, think wisely: Optimizing thinking dynamics for large reasoning models. *arXiv preprint arXiv:2505.21765*, 2025.
- [5] Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.
- [6] Qiguang Chen, Libo Qin, Jiaqi Wang, Jingxuan Zhou, and Wanxiang Che. Unlocking the capabilities of thought: A reasoning boundary framework to quantify and optimize chain-of-thought. *Advances in Neural Information Processing Systems*, 37:54872–54904, 2024.
- [7] Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. M3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221, 2024.
- [8] Qiguang Chen, Libo Qin, Jinhao Liu, Yue Liao, Jiaqi Wang, Jingxuan Zhou, and Wanxiang Che. Rbf++: Quantifying and optimizing reasoning boundaries across measurable and unmeasurable capabilities for chain-of-thought reasoning. *arXiv preprint arXiv:2505.13307*, 2025.
- [9] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*, 2025.
- [10] Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiaqi Wang, Mengkang Hu, Zhi Chen, Wanxiang Che, and Ting Liu. Ecm: A unified electronic circuit model for explaining the emergence of in-context learning and chain-of-thought in large language model. *arXiv preprint arXiv:2502.03325*, 2025.

- [11] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for $2 + 3 = ?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.
- [12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- [13] Yingqian Cui, Pengfei He, Jingying Zeng, Hui Liu, Xianfeng Tang, Zhenwei Dai, Yan Han, Chen Luo, Jing Huang, Zhen Li, Suhang Wang, Yue Xing, Jiliang Tang, and Qi He. Stepwise perplexity-guided refinement for efficient chain-of-thought reasoning in large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18581–18597, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. URL <https://aclanthology.org/2025.findings-acl.956/>.
- [14] Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. Hybrid llm: Cost-efficient and quality-aware query routing, 2024. URL <https://arxiv.org/abs/2404.14618>.
- [15] Yifu Ding, Wentao Jiang, Shunyu Liu, Yongcheng Jing, Jinyang Guo, Yingjie Wang, Jing Zhang, Zengmao Wang, Ziwei Liu, Bo Du, Xianglong Liu, and Dacheng Tao. Dynamic parallel tree search for efficient llm reasoning, 2025. URL <https://arxiv.org/abs/2502.16235>.
- [16] Sugyeong Eo, Hyeonseok Moon, Evelyn Hayoon Zi, Chanjun Park, and Heuiseok Lim. Debate only when necessary: Adaptive multiagent collaboration for efficient llm reasoning, 2025. URL <https://arxiv.org/abs/2504.05047>.
- [17] Sicheng Feng, Gongfan Fang, Xinyin Ma, and Xinchao Wang. Efficient reasoning models: A survey. *arXiv preprint arXiv:2504.10903*, 2025.
- [18] Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025. URL <https://openreview.net/forum?id=wpK4IMJfdX>.
- [19] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [20] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL <https://aclanthology.org/2024.acl-long.211/>.
- [21] Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

- [22] Mengkang Hu, Tianxing Chen, Yude Zou, Yuheng Lei, Qiguang Chen, Ming Li, Yao Mu, Hongyuan Zhang, Wenqi Shao, and Ping Luo. Text2world: Benchmarking large language models for symbolic world model generation. *arXiv preprint arXiv:2502.13092*, 2025.
- [23] Mengkang Hu, Yuhang Zhou, Wendong Fan, Yuzhou Nie, Bowei Xia, Tao Sun, Ziyu Ye, Zhaoxuan Jin, Yingru Li, Qiguang Chen, et al. Owl: Optimized workforce learning for general multi-agent assistance in real-world task automation. *arXiv preprint arXiv:2505.23885*, 2025.
- [24] Jinyang Huang, Xiachong Feng, Qiguang Chen, Hanjie Zhao, Zihui Cheng, Jiesong Bai, Jingxuan Zhou, Min Li, and Libo Qin. Mldebugging: Towards benchmarking code debugging across multi-library scenarios. *arXiv preprint arXiv:2506.13824*, 2025.
- [25] Shijue Huang, Hongru Wang, Wanjuan Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R Fung. Adactrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025.
- [26] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Alexander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [27] Yiyang Ji, Haoran Chen, Qiguang Chen, Chengyue Wu, Libo Qin, and Wanxiang Che. Mpcc: A novel benchmark for multimodal planning with complex constraints in multimodal large language models. *arXiv preprint arXiv:2507.23382*, 2025.
- [28] Zhong-Zhi Li, Xiao Liang, Zihao Tang, Lei Ji, Peijie Wang, Haotian Xu, Xing W, Haizhen Huang, Weiwei Deng, Ying Nian Wu, Yeyun Gong, Zhijiang Guo, Xiao Liu, Fei Yin, and Cheng-Lin Liu. Tl;dr: Too long, do re-weighting for efficient llm reasoning compression, 2025. URL <https://arxiv.org/abs/2506.02678>.
- [29] Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, et al. From system 1 to system 2: A survey of reasoning large language models. *arXiv preprint arXiv:2502.17419*, 2025.
- [30] Guosheng Liang, Longguang Zhong, Ziyi Yang, and Xiaojun Quan. Thinkswitcher: When to think hard, when to think fast. *arXiv preprint arXiv:2505.14183*, 2025.
- [31] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.
- [32] Zehui Ling, Deshu Chen, Hongwei Zhang, Yifeng Jiao, Xin Guo, and Yuan Cheng. Fast on the easy, deep on the hard: Efficient reasoning via powered length penalty. *arXiv preprint arXiv:2506.10446*, 2025.
- [33] Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. Adacot: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning, 2025. URL <https://arxiv.org/abs/2505.11896>.
- [34] Jinghui Lu, Haiyang Yu, Siliang Xu, Shiwei Ran, Guozhi Tang, Siqi Wang, Bin Shan, Teng Fu, Hao Feng, Jingqun Tang, Han Wang, and Can Huang. Prolonged reasoning is not all you need: Certainty-based adaptive routing for efficient llm/mlm reasoning, 2025. URL <https://arxiv.org/abs/2505.15154>.

- [35] Feng Luo, Yu-Neng Chuang, Guanchu Wang, Hoang Anh Duy Le, Shaochen Zhong, Hongyi Liu, Jiayi Yuan, Yang Sui, Vladimir Braverman, Vipin Chaudhary, and Xia Hu. Autol2s: Auto long-short reasoning for efficient large language models, 2025. URL <https://arxiv.org/abs/2505.22662>.
- [36] Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning, 2025. URL <https://arxiv.org/abs/2501.12570>.
- [37] Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, et al. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *arXiv preprint arXiv:2412.09413*, 2024.
- [38] Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. Routellm: Learning to route llms with preference data, 2025. URL <https://arxiv.org/abs/2406.18665>.
- [39] Jiabao Pan, Yan Zhang, Chen Zhang, Zuozhu Liu, Hongwei Wang, and Haizhou Li. DynaThink: Fast or slow? a dynamic decision-making framework for large language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 14686–14695, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.814. URL <https://aclanthology.org/2024.emnlp-main.814/>.
- [40] Ziqing Qiao, Yongheng Deng, Jiali Zeng, Dong Wang, Lai Wei, Fandong Meng, Jie Zhou, Ju Ren, and Yaoyue Zhang. Concise: Confidence-guided compression in step-by-step efficient reasoning. *arXiv preprint arXiv:2505.04881*, 2025.
- [41] Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, 2023.
- [42] Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*, 2024.
- [43] Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-tuning, 2025. URL <https://arxiv.org/abs/2503.07572>.
- [44] Swarnadeep Saha, Archiki Prasad, Justin Chih-Yao Chen, Peter Hase, Elias Stengel-Eskin, and Mohit Bansal. System-1.x: Learning to balance fast and slow planning with language models, 2025. URL <https://arxiv.org/abs/2407.14414>.
- [45] Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *arXiv preprint arXiv:2503.04472*, 2025.
- [46] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

- [47] Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning, 2025. URL <https://arxiv.org/abs/2504.05520>.
- [48] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.
- [49] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- [50] Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage rl. *arXiv preprint arXiv:2505.10832*, 2025.
- [51] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [52] Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. 2025. URL <https://openreview.net/forum?id=W8dxn7hBkO>.
- [53] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.
- [54] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
- [55] Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Zheng Lin, Li Cao, and Weiping Wang. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*, 2025.
- [56] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms, 2025. URL <https://arxiv.org/abs/2502.03373>.
- [57] Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think, 2025. URL <https://arxiv.org/abs/2505.13417>.
- [58] Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenye Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.
- [59] Shengjia Zhang, Junjie Wu, Jiawei Chen, Changwang Zhang, Xingyu Lou, Wangchunshu Zhou, Sheng Zhou, Can Wang, and Jun Wang. Othink-r1: Intrinsic fast/slow thinking mode switching for over-reasoning mitigation, 2025.

Appendix

A. Extreme Model Result

Model	GSM8K			MATH500			AIME24			AMC			OlymBench			AIME25		
	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF	ACC	LEN	EFF
DR. SAF-Ext	86.6	88	98.41	84.1	759	11.08	41.3	4,002	1.03	75.0	1,581	4.74	55.10	1,576	3.50	26.2	3,998	0.66

Table 3: Performance of the extreme model on six mathematical benchmarks. ACC is accuracy (%), LEN is average response length (tokens), and EFF is token efficiency (ACC/LEN).

Here we present the accuracy, average response token length, and token efficiency of the extreme DR. SAF variant (DR. SAF-Ext) after further training on six test sets.

B. Proof and Analysis

We present a mathematical proof demonstrating that the dynamic difficulty-aware length reward algorithm attains superior expected accuracy $\mathbb{E}[\text{Acc}]$ compared to a static length reward algorithm, under the constraint of identical average response length $\mathbb{E}[\ell]$ on a test set. Both algorithms commence from an equivalent base model, sharing initial response lengths and accuracies. The proof proceeds by delineating assumptions, defining the algorithms, establishing accuracy preservation in the dynamic case, quantifying accuracy degradation in the static case, and concluding with a comparative analysis.

B.1. Assumptions and Initial Setup

Problem Distribution The test set consists of a fraction p of “easy” problems (within the Completely Feasible Reasoning Boundary, CFRB, which can be mastered by the model), and $1 - p$ of “hard” problems (within the Partially Feasible Reasoning Boundary, PFRB, which require more complex reasoning), following Chen et al. [6, 10]. Here we take $0 < p < 1$ to capture realistic distributional diversity.

Initial State (Pre-Training) Given a question set \mathcal{Q} , both algorithms before RL have average response length $\mathbb{E}[\ell_0] = L_0$ and average accuracy $\mathbb{E}[\text{Acc}_0] = A_0$. For questions in CFRB, the initial average response length $\mathbb{E}[\ell_0^{\text{CFRB}}] = L_0$ and accuracy $\mathbb{E}[A_0^{\text{CFRB}}] = A_{\text{high}} > A_0$; for questions in PFRB, response length $\ell_0^{\text{PFRB}} = L_0$ and $\mathbb{E}[A_0^{\text{PFRB}}] = A_{\text{low}} < A_0$. Therefore, $A_0 = pA_{\text{high}} + (1 - p)A_{\text{low}}$.

Efficiency Objective The purpose is to achieve an overall shorter average response $\mathbb{E}[\ell] = L < L_0$ via compression, while maintaining accuracy as much as possible. More specifically, for CFRB problems, efficiency can be pushed to a minimum length $\ell_{\min} < L_0$ without loss of accuracy ($\mathbb{E}[A_*^{\text{CFRB}}] \approx A_{\text{high}}$); while for PFRB problems, compressing below a threshold ℓ_{\max} (with $\ell_{\min} < \ell_{\max} \leq L_0$) leads to accuracy degradation. For analytical tractability, assume this degradation is proportional to the excess compression, i.e., $\Delta_{\text{Acc}} = -\gamma(\ell_{\max} - \ell)$ for some $\gamma > 0$.

B.2. Algorithm Definitions

Traditional Static Algorithm: A uniform token-level length penalty $R_{\text{Len}}^{\text{static}} = -\beta \cdot \ell$ ($\beta > 0$) is imposed across all problems, and RL policy updates are driven solely by the total reward, encouraging indiscriminate response shortening.

DR. SAF: For each test problem x , a group of responses $\mathcal{Y} = \{y_1, \dots, y_k\}$ is sampled via Group Relative Policy Optimization (GRPO). The group-level accuracy is $\text{Acc}(\mathcal{Y}|x) = \frac{|\mathcal{C}|}{k}$, where \mathcal{C} is the set of correct responses. If $\text{Acc}(\mathcal{Y}|x) > 0.9$, the instance is classified as CFRB; responses in the shorter half of \mathcal{C} (below group median length), denoted $\mathcal{C}_{\text{short}}$, get a positive compression reward $R_{\text{Len}} = \delta_{\text{comp}} > 0$. Otherwise, the question is classified as PFRB and no compression reward is given. Policy updates focus on boosting the probability of the correct and short $\mathcal{C}_{\text{short}}$ responses.

B.3. Proof of Adaptive Length Management

We provide a rigorous proof that the adaptive (difficulty-aware) DR. SAF algorithm preserves accuracy for simple problems (CFRB) while achieving significant length compression, thanks to selective compression and a regret mechanism. In contrast, static length penalty causes accuracy loss for hard problems (PFRB) under the same average length constraint.

B.3.1. Accuracy Degradation in Static Algorithm

The static algorithm enforces a fixed degree of compression across all problems, targeting average length $L < L_0$ (with $\delta = L_0 L$). As a result:

For CFRB (“easy”) problems, accuracy remains at A_{high} (since these problems tolerate some compression without loss). For PFRB (“hard”) problems, forced compression may reduce length below the threshold ℓ_{max} , at which accuracy begins to degrade. Specifically, if length drops below ℓ_{max} , the expected accuracy change is:

$$\Delta_{\text{Acc}} = -\gamma [\ell_{\text{max}}(L_0 \delta')] < 0, \quad (9)$$

where $\delta' \geq \delta$ is the effective reduction on hard problems, and $\gamma > 0$ reflects degradation strength.

Therefore, total expected accuracy is:

$$\mathbb{E}[\text{Acc}^{\text{static}}] = pA_{\text{high}} + (1 - p) [A_{\text{low}} + \Delta_{\text{Acc}}] \quad (10)$$

$$= A_0 + (1 - p)\Delta_{\text{Acc}} < A_0 \quad (11)$$

That is, while static length penalty may reduce average response length, it “blindly” compresses even hard cases, causing reliability loss on PFRB.

B.3.2. Proof of Accuracy Preservation in Adaptive (DR. SAF) Algorithm

The dynamic algorithm employs two core principles:

- **Selective Compression on High-Accuracy Problems:** Only problems classified as CFRB (i.e., whenever the sampled group accuracy $\text{Acc}(\mathcal{Y}|x) > 0.9$) are considered for compression, confirming the model’s mastery.
- **Preference for Short Correct Responses:** Compression rewards are targeted exclusively at the subset $\mathcal{C}_{\text{short}}$, the shorter portion of correct group members. This subset (at least 45% of group samples) is inherently correct, ensuring feasibility of short solutions.

After each update:

Generalization and Iteration: Policy iteration boosts the probability of short correct outputs on future similar (easy) questions, due to reinforcement on $\mathcal{C}_{\text{short}}$. For a new instance of similar difficulty x' , the

probability that a rolled-out response is both short and correct increases:

$$P(\text{short \& correct} | x') \geq P_{\text{short-correct}} + \epsilon, \text{ with } \epsilon > 0$$

Thus, group accuracy remains > 0.9 .

Regret Mechanism Guards Against Over-Compression: If, for any problem, group accuracy under compression dips to ≤ 0.9 , the problem is reclassified as PFRB and all length rewards are suspended (i.e., learning targets only correctness, not length). This automatically halts and reverses excessive compression, allowing accuracy to recover; if subsequent iterations restore accuracy above 0.9, controlled compression resumes. This feedback prevents any sustained accuracy loss.

No Compression Risk for Hard Problems: For PFRB cases (i.e., those not reliably above the 0.9 group-accuracy threshold), compression is not incentivized and lengths remain essentially unchanged ($\ell_{\text{PFRB}}^{\text{dynamic}} \approx L_0$), with accuracy maintained at A_{low} .

Formal Result: Strict Preservation on Simple Cases By design, the adaptive algorithm’s group-accuracy filter and “regret” feedback guarantee that

$$\mathbb{E}[\text{Acc}_{\text{CFRB}}^{\text{dynamic}}] \geq \min(\text{Acc}(\mathcal{Y}|x)) > 0.9 \approx A_{\text{high}} \quad (12)$$

with length as short as possible subject to this constraint.

Unified Analysis and Aggregate Accuracy Let $\mathbb{E}[\ell_{\text{CFRB}}^{\text{dynamic}}]$ denote the compressed average length for CFRB (comparable or much smaller than in the static case). The overall expected accuracy under adaptive management is

$$\mathbb{E}[\text{Acc}^{\text{dynamic}}] = p \mathbb{E}[\text{Acc}_{\text{CFRB}}^{\text{dynamic}}] + (1 - p)A_{\text{low}} \geq A_0 \quad (13)$$

The global accuracy benefit relative to static is

$$\mathbb{E}[\text{Acc}^{\text{dynamic}}] - \mathbb{E}[\text{Acc}^{\text{static}}] = -(1 - p)\Delta_{\text{Acc}} > 0 \quad (14)$$

Conclusion Dynamic length management (DR, SAF) strictly preserves accuracy for easy (CFRB) problems, shrinking response length to the minimal feasible value without risk of degradation, while automatically suspending compression on hard (PFRB) problems. In contrast, static compression always sacrifices PFRB accuracy to achieve the same average response length. This guarantees that dynamic adaptive management is fundamentally more robust and efficient than static penalization.

B.4. Proof and Analysis of the Boundary Preservation Mechanism

We now give a systematic proof for the Boundary Preservation Mechanism, i.e., the use of truncated mean normalization in the advantage function. This proof covers the algorithmic rationale, the risk of boundary collapse, mathematical guarantees, and comparative stability. All analysis is based on the defined reward framework and boundary/correctness conditions.

B.4.1. Proof Overview

The Boundary Preservation Mechanism modifies the policy advantage by introducing a truncated mean $\mu_{\mathcal{R}}^{\text{trunc}}$, so that all correct responses $y_i \in \mathcal{C}$ have $\mathcal{A}_{\text{Pre}}(y_i|x) \geq 0$. This ensures correct responses are never assigned negative advantages (which would otherwise lead to their probability collapsing), thereby maintaining a non-empty solution set. The proof covers: background, necessity, theoretical guarantee, stability, and comparative performance.

Assumptions and Algorithm Definition Given input x , k sampled responses $\mathcal{Y} = \{y_1, \dots, y_k\}$. The set of correct responses is $\mathcal{C} \subseteq \mathcal{Y}$, of size $m \leq k$. The effective reward for y_i is

$$R_{\text{Eff}}(y_i|x) = R_{\text{Acc}}(y_i|x) + R_{\text{Len}}(y_i|x) + R_{\text{Aware}}(y_i|x). \quad (15)$$

Rewards are aggregated as $\mathcal{R}_{\text{Eff}} = \{R_{\text{Eff}}(y_i|x)\}$. The mean is $\mu_{\mathcal{R}}$, variance $\sigma_{\mathcal{R}}$.

By definition, CFRB responses must satisfy all criteria (accuracy, length, awareness), PFRB only correctness. Boundary collapse means correct responses receive negative advantage and vanish from the policy support.

Under initial policy π_0 , the reward mean for correct is higher than for incorrect, i.e. $\mathbb{E}[R_{\text{Eff}}|y \in \mathcal{C}] > \mathbb{E}[R_{\text{Eff}}|y \notin \mathcal{C}]$, and reward distribution is approximately normal $\mathcal{N}(\mu, \sigma^2)$.

The vanilla GRPO computes

$$A_{\text{vanilla}}(y_i|x) = \frac{R_{\text{Eff}}(y_i|x)\mu_{\mathcal{R}}}{\sigma_{\mathcal{R}}}, \quad (16)$$

and updates $\nabla \log \pi(y|x) \propto A_{\text{vanilla}}(y|x)$.

The proposed mechanism sets

$$\mu_{\mathcal{R}}^{\text{trunc}} = \min(\mu_{\mathcal{R}}, \min_{y_i \in \mathcal{C}} R_{\text{Eff}}(y_i|x)), \quad (17)$$

and

$$\mathcal{A}_{\text{Pre}}(y_i|x) = \frac{R_{\text{Eff}}(y_i|x)\mu_{\mathcal{R}}^{\text{trunc}}}{\sigma_{\mathcal{R}}}, \quad (18)$$

so that all $y_i \in \mathcal{C}$ get non-negative advantages. Policy is updated accordingly.

B.4.2. Necessity Analysis for Boundary Collapse

In the vanilla scheme, long correct responses (especially in PFRB) might get $R_{\text{Eff}} < \mu_{\mathcal{R}}$ due to large length penalty, causing negative advantage and their probability to go down rapidly:

$$P(A_{\text{vanilla}}(y_i|x) < 0) = P(R_{\text{Eff}}(y_i|x) < \mu_{\mathcal{R}}) \approx \Phi\left(\frac{\mu_{\mathcal{R}}\mathbb{E}[R_{\text{Eff}}|\mathcal{C}]}{\sigma_{\mathcal{R}}}\right), \quad (19)$$

where Φ is the standard normal CDF. Strong penalties can make this probability > 0.5 . After iterative updates, the policy probability for these responses

$$\pi_t(y \in \mathcal{C}|x) \approx \pi_{t-1}e^{-\lambda|\bar{A}_{\text{neg}}|}, \quad (20)$$

so after T steps and if $\bar{A}_{\text{neg}} < 0$, $\pi_T \rightarrow 0$ (i.e., collapse occurs). As a result, the overall accuracy drops by $(1 - p)\gamma$ in expectation, matching ΔAcc above and causing instability.

This analysis clarifies that without truncation, probability mass for some correct responses vanishes proportionally to penalty strength and PFRB proportion. Truncation is thus necessary.

B.4.3. Mathematical Guarantees of Truncated Advantages

Truncated mean ensures that for all $y_i \in \mathcal{C}$,

$$\mathcal{A}_{\text{Pre}}(y_i|x) = \frac{R_{\text{Eff}}(y_i|x)\mu_{\mathcal{R}}^{\text{trunc}}}{\sigma_{\mathcal{R}}} \geq 0. \quad (21)$$

Thus, probability of negative advantage for correct responses is zero. Hence the policy for correct responses is never suppressed, so the correct boundary persists across training.

After T training steps, the probability of boundary preservation is at least $1 - e^{-T \min \mathcal{A}_{\text{Pre}}}$, which tends to 1 as T increases.

B.4.4. Overall Stability Proof

Jensen’s inequality tells us that truncating the mean increases the expected advantage for correct answers:

$$\mathbb{E}[\mathcal{A}_{\text{Pre}}|\mathcal{C}] \geq \frac{\mathbb{E}[R_{\text{Eff}}|\mathcal{C}]\mu_{\mathcal{R}}^{\text{trunc}}}{\sigma_{\mathcal{R}}} > \mathbb{E}[A_{\text{vanilla}}|\mathcal{C}]. \quad (22)$$

The policy gradient is thus more stably oriented.

Oscillation is reduced, since extremely negative gradients are eliminated. As a Lyapunov stability argument, define $V_t = \sum_{y \in \mathcal{C}} \log \pi_t(y|x)$. Then V_t is monotonically non-increasing and bounded. Thus, the policy converges stably rather than oscillating.

In summary, expected accuracy remains at least A_0 after compression.

B.4.5. Comparative Analysis and Conclusion

Relative to vanilla GRPO, the boundary preservation mechanism reduces accuracy collapse probability from $(1 - p)P(\Delta_{\text{Acc}} < 0)$ to 0, increases the expected advantage for correct answers, and guarantees stability. Numerically, boundary existence probability multiplies by a factor at least $e^{T(\mathcal{A}_{\text{Pre}} - \mathcal{A}_{\text{vanilla}})} > 1$. Thus, this algorithm supports efficient compression while protecting accuracy and convergence.

C. Experimental Setup

In our experiments, we utilize the VERL training framework to implement the proposed methodology. The training is conducted using the PPO trainer with GRPO as the advantage estimator. Key parameters for data handling, model configuration, rollout settings, and trainer options are detailed below. These parameters are optimized for efficient compression while maintaining model accuracy on mathematical reasoning tasks. Validation-specific parameters are separated into a dedicated table for clarity. We found that a maximum response length of 16384 may truncate some high-quality long responses, while 32768 is too slow for training and likely to cause GPU OOM. To speed up training and prevent OOM issues, while ensuring that longer high-quality responses can be sampled, we set the maximum response length to a balanced value of 22000. In testing the Qwen3 and Qwen2.5 undistilled DeepSeek models without RL training, we set the maximum response length to 32768. Because the AIME and AMC datasets are relatively small, we conduct evaluations on each dataset multiple times, ten runs for AIME and two runs for AMC, and report the mean accuracy across these repetitions.

Parameter	Value
val temperature	0.6
val do sample	True
val top_k	45
val batch size	512

Table 4: Validation Parameters of the VERL Training Framework.

Parameter	Value
advantage estimator	grpo
train batch size	8 (8×12)
max prompt length	2048
max response length	22000
use remove padding	True
ppo mini batch size	8
ppo micro batch size per gpu	1
log prob micro batch size per gpu	1
use kl loss	False
entropy coeff	0.001
enable gradient checkpointing	True
fsdp param offload	True
fsdp optimize offload	True
model parallel size	2
actor rollout engine	vllm
rollout num per question	12
num nodes	1
gpus per node	8 (A100-80G)
gpu memory utilization	0.8

Table 5: Main Parameters of the VERL Training Framework.